Machine Learning 1

Lecture 11.5 - Kernel Methods
Support Vector Machines - Kernel SVM

Erik Bekkers

(Bishop 7.1.0)

Slide credits: Rianne van den Berg

Image credit: Kirillm | Getty Images

Maximum Margin Classifier

- Maximizing the margin:

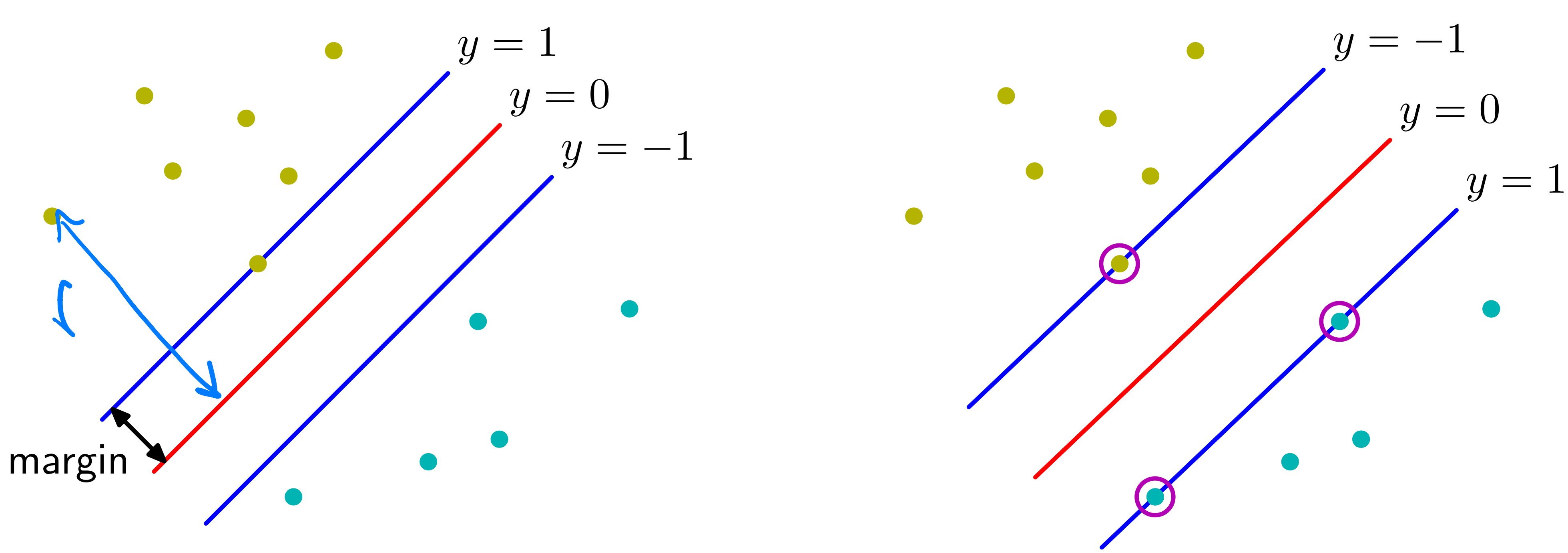
$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \textit{ subject to } N \textit{ constraints } t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

- We decided to “calibrate” $\mathbf{w}$ s.t. for the nearest point $t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$

- Then the size of the margin is given by $\dfrac{1}{\|\mathbf{w}\|}$

$r = \dfrac{t_n(\underline{w}^T x_n + b)}{\|\underline{w}\|}$

- And for all data points we have $t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$



Maximum Margin Classifier

- Maximizing the margin:

$$\underset{\mathbf{w},b}{\arg\min}\,\frac{1}{2}\|\mathbf{w}\|^2 \textit{ subject to } N \textit{ constraints } t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

- Primal Lagrangian function:

Lagrange multiplier

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n\{\underbrace{t_n(\mathbf{w}^T\mathbf{x}_n + b)}_{g(w)} - 1\}$$

$f(w)$ $-$ a_n

- With KKT conditions:

(primal feasibility)	$t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0$	for	$n = 1,\ldots,N$
(dual feasibility)	$a_n \geq 0$	for	$n = 1,\ldots,N$
(complimentary slackness)	$a_n(t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1) = 0$	for	$n = 1,\ldots,N$

- Dual Lagrangian obtained via (stationarity conditions) $\dfrac{\partial L}{\partial \mathbf{w}} = 0, \quad \dfrac{\partial L}{\partial b} = 0$

$$\tilde{L}(\mathbf{a}) = \min_{\mathbf{x},b} L(\mathbf{x}, b, \mathbf{a})$$

- **Solution**: $\mathbf{a}^* = \underset{\mathbf{a}}{\arg\max}\,\tilde{L}(\mathbf{a}) \longrightarrow \mathbf{w}^*, b^* = \underset{\mathbf{w},b}{\arg\min}\,L(\mathbf{w}, b, \mathbf{a}^*)$

Maximum Margin Classifier

- Primal Lagrangian function:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1\}$$

with Langrange multipliers: $a_n \geq 0$ for $n = 1, \ldots, N$

- First step towards dual Langrangian: obtain stationarity conditions

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w}^T - \sum_{n=1}^{N} a_n t_n \mathbf{x}_n^T = 0 \quad \rightarrow \quad \boxed{\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n}$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} a_n t_n = 0 \quad \rightarrow \quad \boxed{\sum_{n=1}^{N} a_n t_n = 0}$$

- Eliminate $\mathbf{w}$ and b from L then gives the dual representation!

Maximum Margin Classifier

- Stationarity conditions:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w}^T - \sum_{n=1}^{N} a_n t_n \mathbf{x}_n^T = 0 \quad \rightarrow \quad \boxed{\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n}$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} a_n t_n = 0 \quad \rightarrow \quad \boxed{\sum_{n=1}^{N} a_n t_n = 0}$$

- Eliminate $\mathbf{w}$ and b from L then gives the dual representation!
  - Primal: $L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1\},$ with $a_n \geq 0$ for $n = 1,\ldots,N$
  - Dual: $\tilde{L}(\mathbf{a}) = w^T\left(\frac{1}{2} w - \boxed{\sum_{n=1}^{N} a_n t_n x_n}\right) - \sum_{n=1}^{N} a_n t_n b + \sum_{n=1}^{N} a_n$

$$= -\frac{1}{2} w^T w - b \underbrace{\sum_{n=1}^{N} a_n t_n}_{=0} + \sum_{n=1}^{N} a_n$$

$$= \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \mathbf{x}_n^T \mathbf{x}_m$$

with $a_n \geq 0$ for $n = 1,\ldots,N$

and with $\sum_{n=1}^{N} a_n t_n = 0$

Maximum Margin Classifier

- The dual representation of the maximum margin, where we maximize w.r.t. **a**:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \mathbf{x}_n^T \mathbf{x}_m$$

with constraints: $a_n \geq 0$ for $n = 1,\ldots,N$

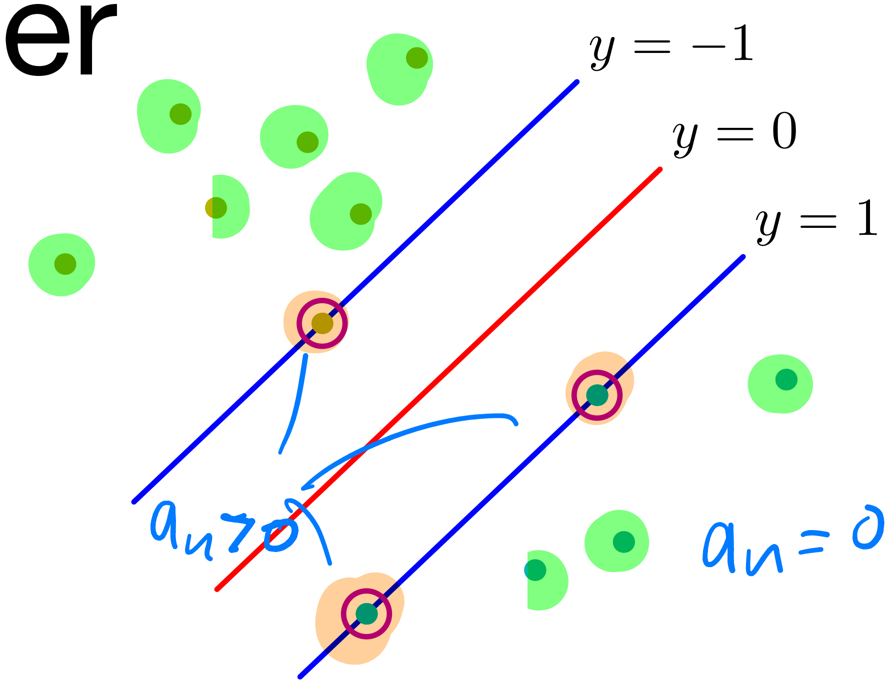
$$\sum_{n=1}^{N} a_n t_n = 0$$

- Apply the **KERNEL TRICK**: replace $\mathbf{x}_n^T\mathbf{x}_m$ with $k(\mathbf{x}_n, \mathbf{x}_m)$

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

- Advantage: can now learn complex nonlinear decision boundaries!

Maximum Margin Classifier



- Prediction of class for datapoint $\mathbf{x}_n$:

$$y(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n + b$$

- Use $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$ so that

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n^T \mathbf{x} + b \xrightarrow{\text{kernel trick!}} y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}_n, \mathbf{x}) + b$$

dual formulation

- Remember the KKT conditions:

(primal feasibility)	$t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 \geq 0$	for	$n = 1, \ldots, N$
(dual feasibility)	$a_n \geq 0$	for	$n = 1, \ldots, N$
(complimentary slackness)	$a_n(t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1) = 0$	for	$n = 1, \ldots, N$

- Support vectors lie on maximum margin hyperplanes

$a_n > 0$	$\rightarrow$	$t_n y(\mathbf{x}_n) = 1$	(support vectors)
$a_n = 0$	$\leftarrow$	$t_n y(\mathbf{x}_n) > 1$	(all other points)

Maximum Margin Classifier

- Prediction of class for datapoint $\mathbf{x}$:

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n^T \mathbf{x} + b \quad \rightarrow \quad y(\mathbf{x}) = \sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}) + b$$

- Find b by using that $t_n y_n(\mathbf{x}) = 1$ if $\mathbf{x}_n$ lies on the margin boundary! ($\mathbf{x}_n$ is a support vector)

- Then $$t_n \left(\sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}_n) + b \right) = 1$$

· t_n on both sides

$(t_n)^2 = 1$

$$\sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}_n) + b = t_n$$

$$\rightarrow \quad b = t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}_n)$$

- More stable to average over all support vectors (depending on optimizer, $\mathbf{a}_n$ may not be perfect)

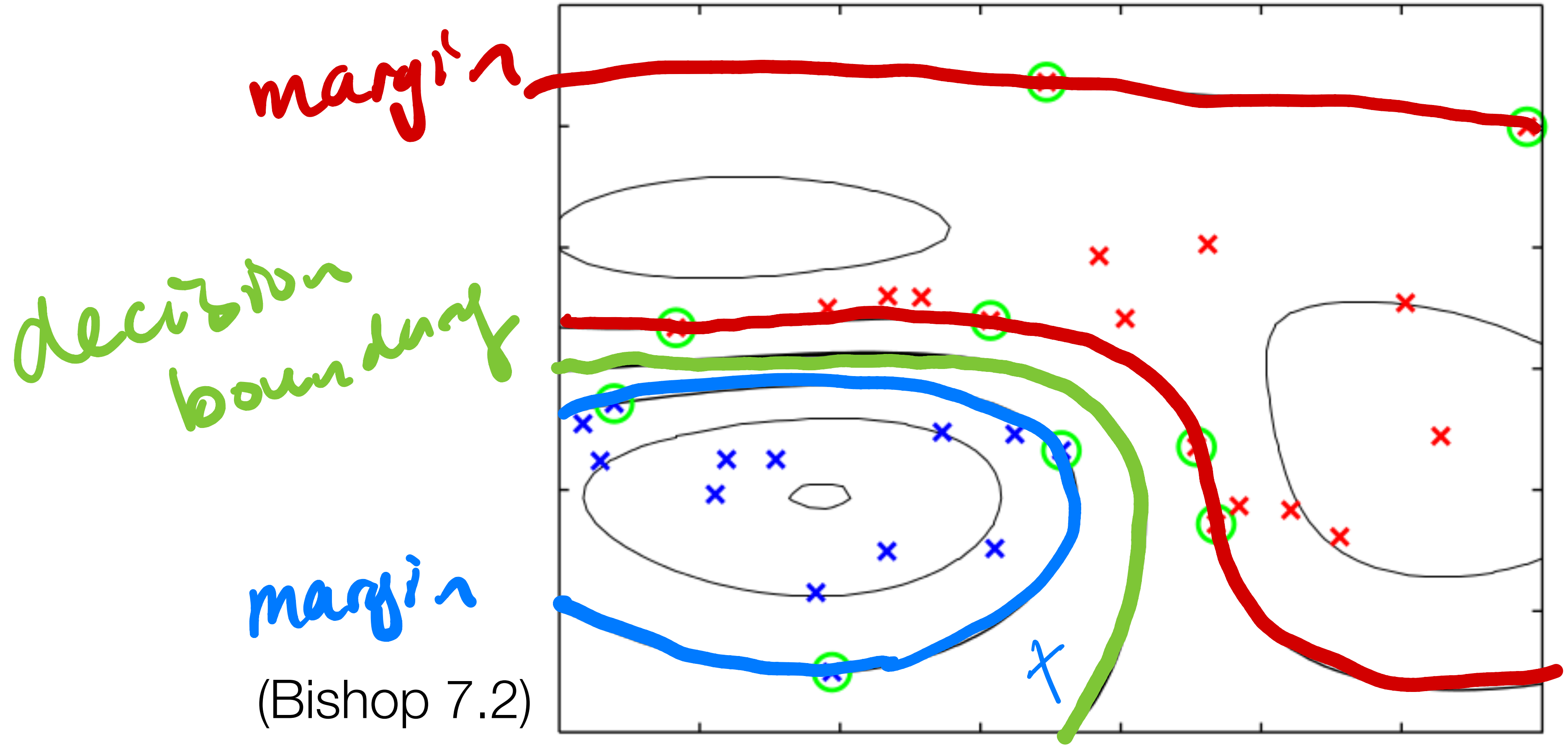
$$\rightarrow \quad b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}_n) \right)$$

Maximum Margin Classifier

- Maximum Margin Classifier with Gaussian Kernel

$$y(\mathbf{x}) = \sum_{m \in S} a_m t_m k(\mathbf{x}_m, \mathbf{x}) + b, \quad \text{with} \quad k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_n - \mathbf{x}_m\|^2\right)$$

- Dataset is not linearly separable
- Nonlinear kernel can still separate the data perfectly!



(Bishop 7.2)